

**Государственное бюджетное профессиональное образовательное
учреждение
Ростовской области
«РОСТОВСКИЙ-НА-ДОНУ КОЛЛЕДЖ СВЯЗИ И ИНФОРМАТИКИ»**

Отчет
по практическому занятию №3
по дисциплине «Основы алгоритмизации и программирования»

Специальность: 09.02.07 «Информационные системы и программирование»

Выполнила
Студентка группы ИС-28
Скворцова Татьяна Александровна
Преподаватель:
Манакова Ольга Петровна

Ростов-на-Дону, 2024

Практическое занятие № 2

Наименование практического занятия: составление программ ветвящейся структуры в IDE PyCharm Community.

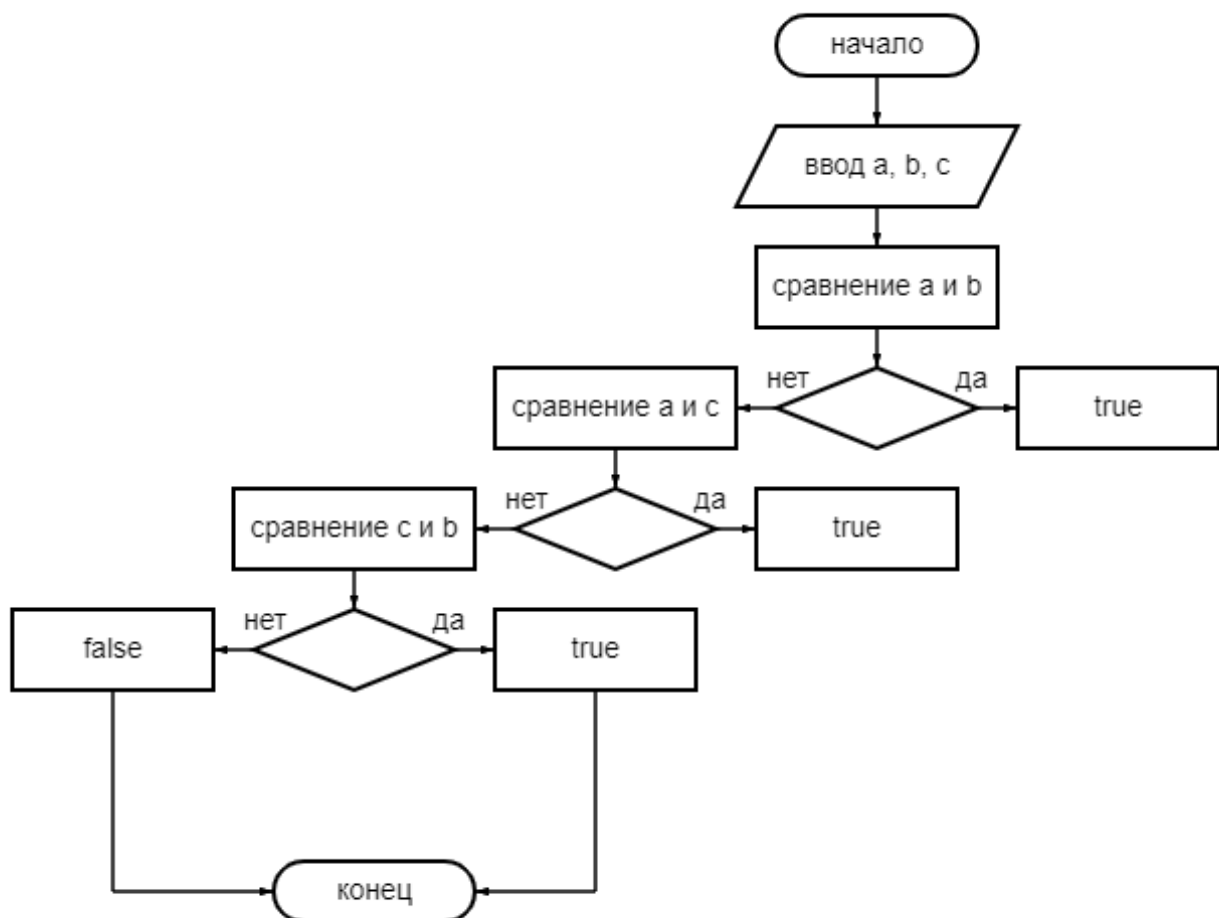
Количество часов: 6.

Цели практического занятия: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ ветвящейся структуры в IDE PyCharm Community.

Задание 1.

Постановка задачи: необходимо разработать программу, которая проверяет истинность высказывания: «Среди трех данных целых чисел есть хотя бы одна пара совпадающих». Программа должна запрашивать у пользователя ввод трех целых чисел и выводить результат проверки. Если хотя бы одна пара чисел совпадает, программа должна выводить True, в противном случае — False.

Блок-схема алгоритма:



Тип алгоритма: разветвляющийся. В процессе выполнения программа принимает решения на основе логических условий. В зависимости от того, совпадают ли числа, происходит один из двух возможных исходов.

Текст программы на Python:

```
# Проверить истинность высказывания: «Среди трех данных целых чисел есть хотя бы одна пара совпадающих»
```

```
try:
```

```
    a = int(input("Введите первое число: "))
```

```
    b = int(input("Введите второе число: "))
```

```
    c = int(input("Введите третье число: "))
```

```
    # Проверка наличия хотя бы одной пары совпадающих чисел
```

```
    if a == b or a == c or b == c:
```

```
        print("Среди трёх данных чисел есть хотя бы одна пара  
совпадающих: ", True)
```

```
    else:
```

```
        print("Среди трёх данных чисел нет ни одной пары совпадающих:  
", False)
```

```
except ValueError:
```

```
    print("Пожалуйста, введите целое число!") # обработка исключений
```

```
    print(f"Полученное число: {result}") # Вывод результата функции  
rotate_number, либо сообщения об ошибке
```

```
    # Сообщение об ошибке на случай, если было введено не число  
(текст, буквы, символы)
```

```
except ValueError:
```

```
    print("Ошибка: Введено не число.")
```

Протокол работы программы:

Введите первое число: 45

Введите второе число: 12

Введите третье число: 45

Среди трёх данных чисел есть хотя бы одна пара совпадающих: True

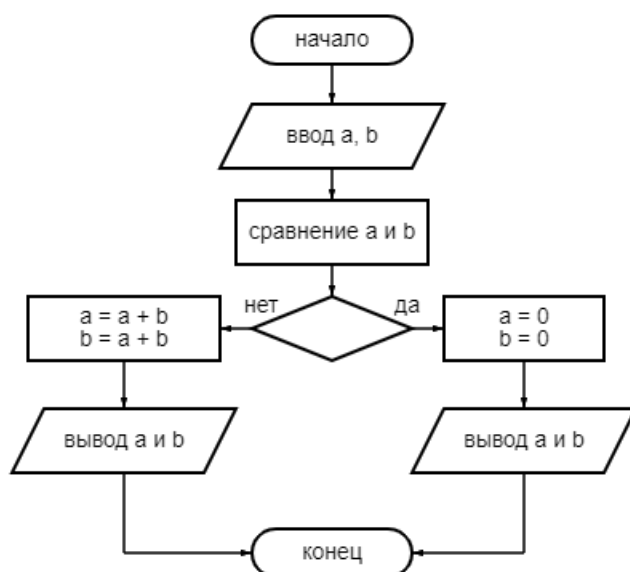
Вывод по заданию 1:

В данном отчете была описана задача по проверке на совпадение целых чисел, представлен алгоритм решения в виде блок-схемы, а также приведен пример кода на Python, реализующий данный алгоритм. Код успешно проверяет, есть ли среди трех введенных чисел хотя бы одна пара совпадающих.

Задание 2.

Постановка задачи: Необходимо разработать программу, которая работает с двумя переменными целого типа: А и В. Если значения переменных не равны, то каждой переменной следует присвоить сумму этих значений. Если же значения равны, переменным нужно присвоить нулевые значения. В конце программа должна вывести новые значения переменных А и В.

Блок-схема алгоритма:



Тип алгоритма: разветвляющийся. В процессе выполнения программа принимает решения на основе логических условий. В зависимости от того, совпадают ли числа, происходит один из двух возможных исходов.

Текст программы на Python:

```
# Даны две переменные целого типа: A и B.
# Если их значения не равны, то присвоить каждой переменной сумму
этих значений, а если равны, то присвоить переменным нулевые
значения.
# Вывести новые значения переменных A и B.

try:
    a = int(input("Введите первое число: "))
    b = int(input("Введите второе число: "))

    if a == b:
        a = 0
        b = 0
    else:
        c = a + b
        a = c
        b = c

    print ("Результат:")
    print("Первое число:", a)
    print("Второе число:", b)

except ValueError:
    print("Пожалуйста, введите целое число!") # обработка исключений
```

Протокол работы программы:

```
Введите первое число: 45
Введите второе число: 45
Результат:
Первое число: 0
Второе число: 0
```

Вывод по заданию 2:

В данном отчете была описана задача, связанная с работой с двумя целыми переменными А и В. Приведен алгоритм решения задачи в виде блок-схемы, а также представлен пример кода на Python, который реализует указанный алгоритм. Код успешно обрабатывает вводимые значения и выводит новые значения переменных в соответствии с заданными условиями.

Вывод по практическому занятию №3:

В процессе выполнения практического занятия выработала навыки составления программ линейной структуры в IDE PyCharm Community. Были использованы языковые конструкции такие как:

1. try - начало блока для обработки исключений.
2. except - блок обработки исключений.
3. int() - функция для преобразования входных данных в целое число.
4. input() - функция для получения ввода от пользователя.
5. if - условная конструкция для проверки условий.
6. or - логический оператор для объединения условий.
7. Print() - функция для вывода данных на экран.
8. True - булево значение, представляющее истину.
9. False - булево значение, представляющее ложь.

Выполнена разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.